# PRODUCT REQUIREMENTS DEVELOPMENT AND MANAGEMENT PROCEDURE

**Approval __original signed on file_______ Date __7/21/06____**
**Associate Director for Operations**

NOTE: USE OF THIS PROCEDURE IS MANDATORY FOR CENTER MANAGEMENT DESIGNATED PROJECTS. THIS PROCEDURE IS BASED ON "BEST PRACTICES" AS TAUGHT IN THE NASA ENGINEERING TRAINING (NET) PROGRAM'S SYSTEM REQUIREMENTS (REQ) CLASS [2]. **USERS SHOULD NOT ENGAGE IN THE ACTIVITIES COVERED BY THIS CP WITHOUT FIRST HAVING COMPLETED THE NET SYSTEMS REQUIREMENTS or SYSTEMS REQUIREMENTS–TEAM TRAINING or EQUIVALENT TRAINING WHICH COVERS THE SAME PHASES AS THIS PROCEDURE.** THE PRIMARY REFERENCE IS THE TEXT *CUSTOMER CENTERED PRODUCTS: CREATING SUCCESSFUL PRODUCTS THROUGH SMART REQUIREMENTS MANAGEMENT*, I. F. HOOKS AND K. A. FARRY, AMACOM (2001), ISBN 0-8144-0568-1. APPLICATION FEEDBACK IS WELCOME AND ANY QUESTIONS OR COMMENTS SHOULD BE DIRECTED TO THE AUTHORS, WHO MAY BE REACHED VIA EMAIL AT rdmp_feedback@larc.nasa.gov. SUPPORTING MATERIAL, REFERENCES, SAMPLES, AND APPLICABLE DOCUMENTS CAN BE FOUND AT URL: http://sw-eng.larc.nasa.gov/docs/requirements_capture_and_management.html

# PRODUCT REQUIREMENTS DEVELOPMENT AND MANAGEMENT

## Objectives:

- to define the process to generate, document, review, baseline, and manage product requirements for a system-of-interest
- to ensure customer and user product requirements are understood and met
- to define a set of complete, consistent, unambiguous, and verifiable product requirements

## Scope:

- This procedure applies to the development of technical product requirements (e.g., System Requirements, Software Requirements, or Interface Requirements). This procedure, LMS-CP-5526, applies to Critical-Control and High-Control projects; Low-Control projects are required to follow LMS-CP-5524. Minimal-Control projects are not required to follow either procedure. For instructions on how to determine whether a project is Critical, High, Low, or Minimal-Control, see Appendix D: Criteria for Procedure Applicability.
- If the product will be obtained under contract, the corresponding Statement of Work requirements are reviewed by following LMS-CP-5523.
- This procedure applies to the top-level systems of interest, but not to the lowest levels, such as component, piece part, or software subroutine. The governing Project Office will determine to what architectural level this procedure should be applied and document the determination in the project's System Engineering Management Plan or equivalent. (For guidance see [1].)
- To fully understand all the technical content of the procedure, users are expected to have been trained in the NASA Engineering Training classes [2] required in the Input/Entry Criteria for Sections 2 through 5.

## Table of Contents

**Section 1: Overview of Procedure** ........ **3**
1.1 How to use this document ........ 3
1.2 Records Generated ........ 3
1.3 Context Diagram ........ 3
**Section 2: Document the Scope (of the system-of-interest)** ........ **4**
**Section 3: Document the Requirements (of the system-of-interest)** ........ **8**
**Section 4: Validate the Requirements (of the system-of-interest)** ........ **11**
**Section 5: Baseline and Manage Requirements (of the system-of-interest)** ........ **14**
**Appendix A: Rules for Writing Good Requirements** ........ **16**
A.1 Use of Correct Terms ........ 16
A.2 Editorial Checklist ........ 16
A.3 Goodness Checklist ........ 16
A.4 Content Review / Inspection Checklist ........ 17
**Appendix B: Formal Review/Inspection Instructions** ........ **19**
B.1 Formal Review Instructions ........ 19
B.2 Formal Inspection Instructions ........ 19
**Appendix C: Acronyms, Definitions, and References** ........ **20**
C.1 Acronyms ........ 20
C.2 Definitions ........ 20
C.3 References ........ 21
**Appendix D: Criteria for Procedure Applicability** ........ **22**

# Section 1: Overview of Procedure

## 1.1 How to use this document

- This procedure assumes the governing Project Office produces the following documents:
  - Project Plan
  - System Engineering Management Plan
  - Risk Management Plan
  - Configuration Management Plan
- Consult the governing Project Office on the following documents:
  - Requirements Management Tool Work Instruction
  - Document Tree
  - Document Templates
  - Glossary
- Refer to the project’s Document Templates for instructions on capturing individual items described below and for organizing Requirements Documents.
- Use the project’s Glossary to ensure consistent use of terminology.
- The context diagram below (Section 1.3) defines, at a high level, the activities of this procedure. Sections 2 through 5 expand the activities shown in the context diagram into flow diagrams and detailed steps.
- Inputs/Entry Criteria indicate the items that must be available and actions that must be completed before beginning the section of the procedure.
- Outputs/Exit Criteria indicate the items that must be available and the actions that must be completed before leaving the section of the procedure.
- Some of the boxes in the flow diagrams are numbered with their associated step, which provides detailed instructions for performing the actions in that box. If no number is given, there are no additional details provided.
- Numbers in brackets, which are embedded in the text, e.g., [2], call out references listed in Appendix C.3.
- Apply the Section 2 through 5 flowcharts recursively to each system-of-interest (e.g., system, flight segment, ground segment, instrument, and spacecraft).
- Text labeled as “GUIDANCE” is informative and not a requirement.

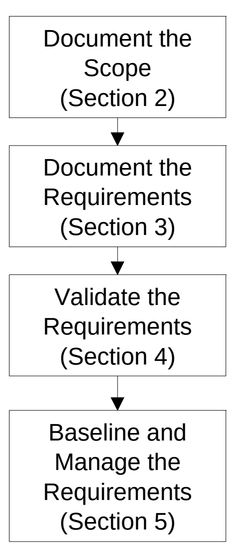
## 1.2 Records Generated

The following records are generated by this procedure:

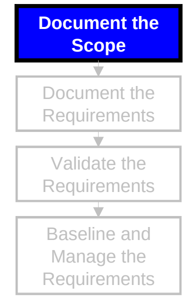
- Scope Documentation
- Requirements Document(s) (e.g., System Requirements Document, Interface Requirements Document (IRD))
- Process Metrics
- Change Request(s) (CRs)
- Evidence that review / inspection issues were resolved

## 1.3 Context Diagram

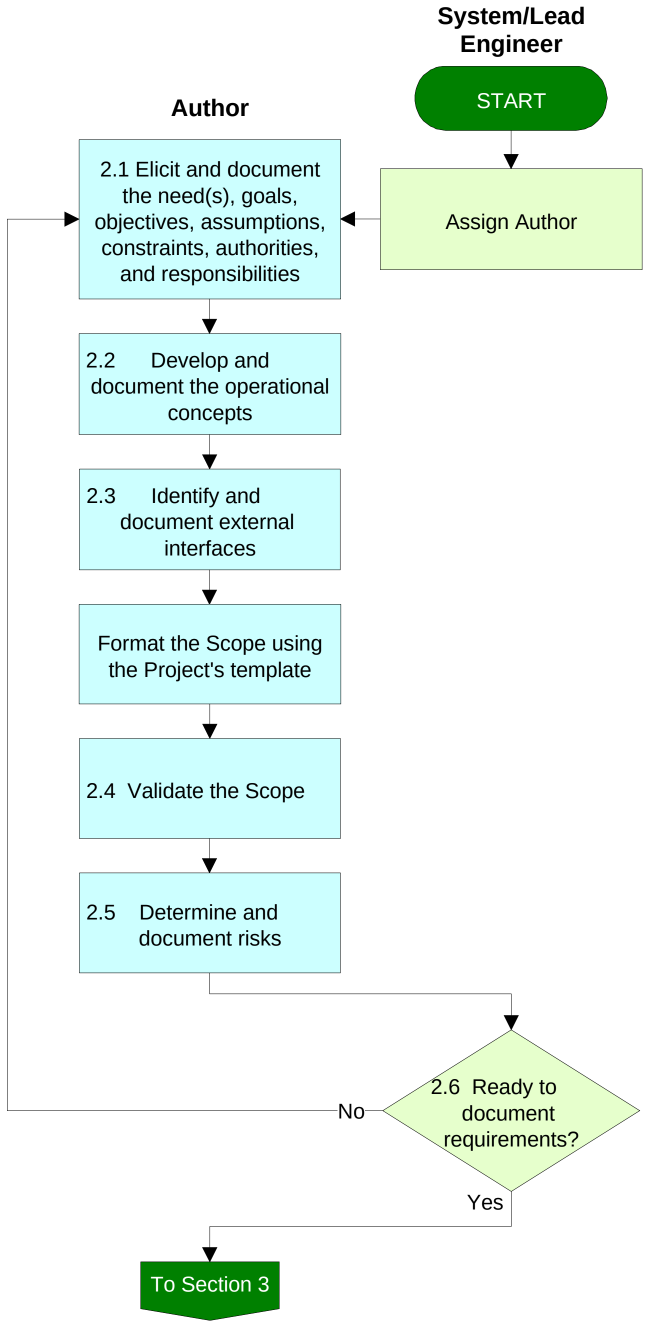
This Product Requirements Development and Management Procedure consists of four major phases shown below.



## Section 2: Document the Scope (of the system-of-interest)




**Instructions:** The actions defined in the boxes below are performed in parallel and iteratively.




**Inputs/Entry Criteria:**

- All stakeholder groups are identified and the names of the representatives are documented.
- System-of-interest parent documents are available i.e., parent Scope, Requirements, IRDs, and Interface Control Documents (ICDs).
- The authors and system/lead engineers have been trained on how to develop and document the Scope, i.e., NET REQ or REQ-T class [2], or equivalent training which covers the same phases as this procedure.

**Outputs/Exit Criteria:**

- Scope documentation is reviewed and approved for a feasible approach that will satisfy the product need.
- Risks have been identified and mitigation plans developed per the project's Risk Management Plan.

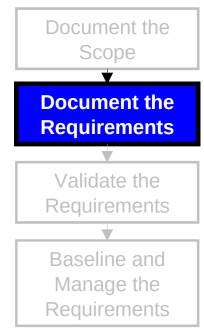
| STEP | ACTION TO BE TAKEN |
|---|---|
| **2.1** | **Elicit and document the need(s), goals, objectives, assumptions, constraints, authorities and responsibilities**<br>a. Become familiar with system-of-interest parent documents.<br>GUIDANCE: The system-of-interest's need, goals, and objectives are often contained in the parent documents. The project need is often found in the announcement of opportunity or proposal. The project's goals and objectives are generally found in the Project Plan or Proposal.<br>b. Document the need(s). Align all stakeholders to one vision of the need through discussions and reviews.<br>GUIDANCE: The 'need' is not a definition of the product or solution. The 'need' explains why the project is developing this product from the stakeholders' point of view (i.e., What problem do the stakeholders want to solve?). The 'need' is not likely to change much during the project. Each stakeholder has an individual perspective, agenda, priorities, constraints, understanding of the environment, alternate solutions, and lessons learned. A common set of expectations/vision/need among stakeholders must be formed by sharing knowledge about these perspectives, ensuring that the expectations are realistic, and obtaining agreement by all stakeholders on the needs. Push for a single need statement, but recognize there are exceptions to this rule. If more than one need is identified, ask if they could be stated as a goal or objective. All stakeholders should agree to a common vision of what the 'need' is before the Scope definition can be completed and a full set of quality requirements can be generated.<br>c. Document the goals that define what must be accomplished to meet the need(s).<br>1. Elicit and document goals from all stakeholders.<br>2. Analyze the goals, ensure alignment with need, obtain missing information, and resolve conflicts.<br>3. Prioritize the goals.<br>d. Document the objectives that define, in measurable terms, how the project plans to meet the goals.<br>GUIDANCE: Objectives are initiatives that implement the goals. The objectives also specify the success criteria (i.e., What is the minimum that the stakeholders expect from the system for it to be successful?).<br>e. Explicitly document all constraints (i.e., external items that cannot be controlled and that must be met, which are identified while defining the scope).<br>f. Document the specific authority and responsibility (e.g., to contractor, project technical lead, customer) for aspects of the products development, identified while defining the scope.<br>g. Explicitly, document all assumptions that are identified while defining the need(s), goals, objectives, constraints, authority, and responsibility.<br>h. Validate assumptions (e.g., obtain confirmation from stakeholders, perform prototype or study to determine feasibility). |
| **2.2** | **Develop and document the operational concepts**<br>a. Develop and document operational concepts and associated scenarios for each operational mode, mission phase (e.g., installation, startup, typical examples of normal and contingency operations, shutdown, and maintenance), and lifecycle phase (e.g., development, integration, test, and validation) from all stakeholders' points of view.<br>GUIDANCE: Operational scenarios are a step-by-step description of how the proposed system should operate and interact with its users and its external interfaces (e.g., other systems). Imagine the operation of the future product and document, from the stakeholder's perspective, the steps of how the end-to-end system will function or be used. Choose scenarios that best fit the needs. For additional guidance on operational concept and scenarios, see IEEE 1362 [3].<br>1. Document 'Nominal' scenarios (i.e., scenarios that cover 'normal' operations and environments). Consider the questions: Who will use the product? Why? Where? When? How? Under what conditions? Use the interface diagram discussed in Step 2.3 while discussing and developing the operational concepts with the stakeholders.<br>2. Document 'Off-Nominal' scenarios (i.e., scenarios to cover abnormal operations and environments). Consider the following: hazards to users, others, the product, other products; potential misuse of the product; extreme conditions.<br>3. Refine the scenarios to cover all interfaces. Cover the following interactions: inputs expected; outputs expected; input does not occur; wrong input occurs; wrong output occurs.<br>b. Validate the scenarios by iterating on the above until all the stakeholders agree on the correctness, completeness, and feasibility of the scenarios. |

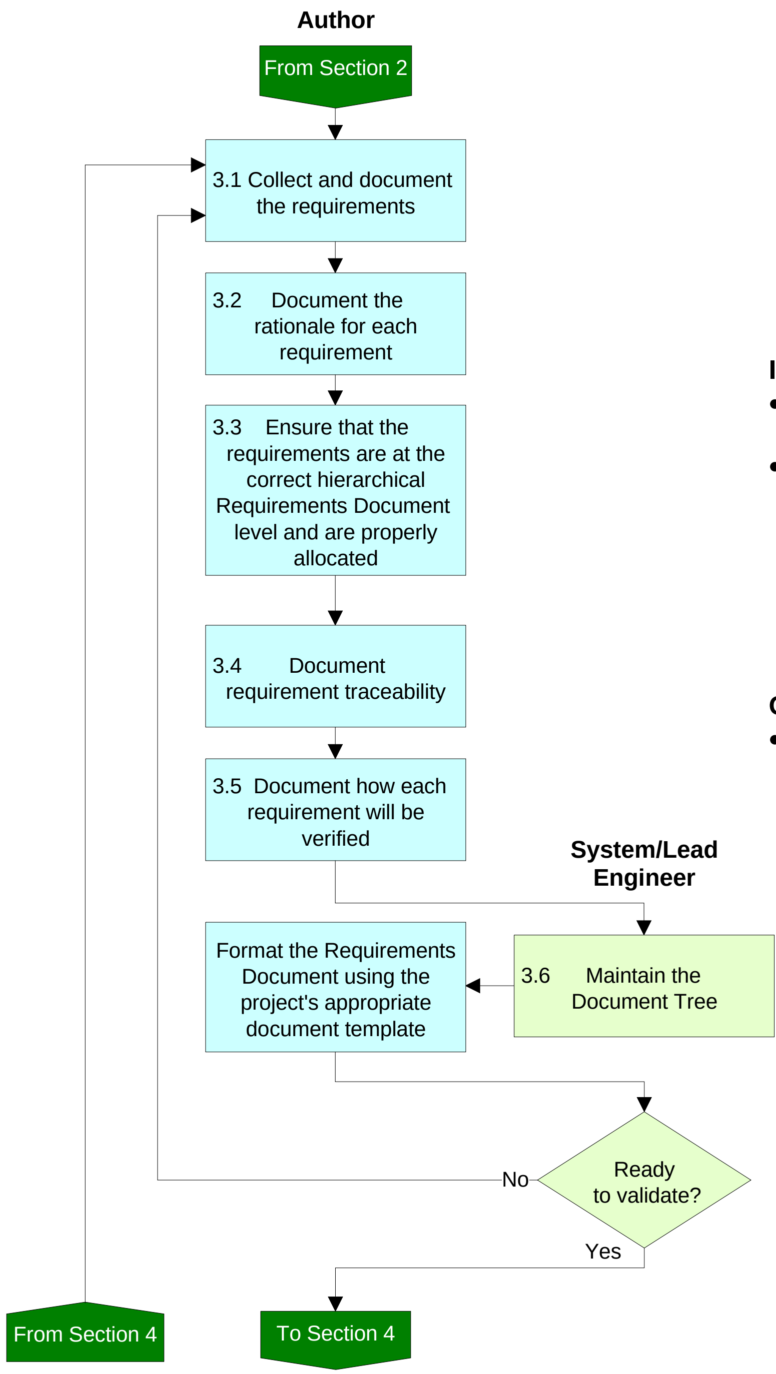
| STEP | ACTION TO BE TAKEN |
|---|---|
| **2.3** | **Identify and document external interfaces**<br>a. Document all external interfaces, including those to enabling systems, which are identified while documenting the need, goals, objectives, and operational concept. Document all the potential interfaces as well (e.g., those for power, structural or physical, data or signal, command or control).<br>GUIDANCE: The external interfaces form the boundaries between the system-of-interest and the rest of the world.<br>b. Ask the following questions about each boundary to the system-of-interest, considering each product lifecycle phase (development through operation and disposal) and document the answer:<br>• What does the product do to/for the world?<br>• What does the world do to/for the product?<br>• What is the worst thing that can happen across this interface?<br>• Is the interface likely to change during the development of the product?<br>• Is this interface likely to change after the product is in use?<br>c. Create the external interface diagram to the system-of-interest. The diagram should depict all the interfaces documented in the steps above.<br>d. Document every industry standard, application programmer's interface, or ICD that exists for the external interfaces.<br>e. Identify IRDs that must be developed for each external interface that does not exist or does not have an ICD.<br>f. Document internal interfaces information, as it is determined.<br>GUIDANCE: Internal interfaces are not addressed in detail at this point. Because dividing a system-of-interest into lower-level systems of interest is a design task, the author should leave a hole in the Requirements Document for these internal interface requirements. The author can go back and fill in this missing information as the design evolves; this means the author may have to submit a change request. An alternative to leaving missing information in the Requirements Document is to develop a separate IRD for each interface; this adds other documents with associated authority and controls.<br>g. Use the following questions to verify interfaces.<br>• Have you identified and documented all product interfaces?<br>• Have you located ICDs for interfaces to existing products?<br>• Have you created a mechanism to monitor interface changes outside your control?<br>• Have you involved people from the other side of the external interface in the interface definition effort?<br>• Have you simplified interfaces as much as possible?<br>• Have you distributed the product interface documentation?<br>• Have you created a mechanism for tracking interfaces through development to ensure that reality matches the documentation?<br>h. Maintain the interface diagram(s). |
| **2.4** | **Validate the Scope**<br>Conduct reviews with representatives from all stakeholder groups to:<br>• Determine validity of the scope<br>• Ensure alignment with system-of-interest parent documents<br>• Identify problems and risks<br>• Find helpful suggestions<br>• Ensure that everyone has the same expectations of the system-of-interest<br>• Ensure scope represents a feasible approach to meeting need<br>• Obtain concurrence of stakeholders<br>GUIDANCE: Once the Scope seems to be fairly stable, consider conducting a Formal Inspection of the documented Scope to eliminate remaining errors, following the Instructional Handbook for Formal Inspection [4]. |

<table>
<tr><th>STEP</th><th>ACTION TO BE TAKEN</th></tr>
<tr><td>2.5</td><td>Determine and document risks<br>Determine and document any remaining risks associated with the recorded Scope following the project’s Risk Management Plan. Use the following questions to help identify risks. A”YES” answer indicates risk.
<ul>
<li>Do we have product boundary questions?</li>
<li>Have we missed or been unable to obtain a key stakeholder input?</li>
<li>Have we missed a product lifecycle phase?</li>
<li>Are there poorly defined or incomplete interfaces?</li>
<li>Are there areas of strong disagreement?</li>
<li>Are there too many unknowns?</li>
<li>Are there assumptions that have not been confirmed with the project or stakeholder personnel?</li>
<li>Are there technical issues?</li>
<li>Are there technology issues?</li>
<li>Are there schedule issues (e.g., overly optimistic)?</li>
<li>Are there cost issues (e.g., budget too lean)?</li>
<li>Are there too many uncertainties?</li>
</ul>
<u>GUIDANCE</u>: Identified risks should be addressed and mitigated according to the project’s Risk Management Plan.</td></tr>
<tr><td>2.6</td><td>Ready to document requirements?<br>Decide if ready to proceed to documenting the requirements.<br><u>GUIDANCE</u>: This is a critical technical and project management decision. To proceed with writing of requirements while there are significant open issues or questions is a risk that could result in incorrect or incomplete requirements and wasted resources.
<table>
<tr><th>If</th><th>Then</th></tr>
<tr><td>High risk</td><td>The requirements writing phase should be postponed until the risk can be reduced. Repeat the associated steps above to reduce the risk.</td></tr>
<tr><td>Low risk</td><td>Put mitigation plan into action and start requirements writing. If the Scope is rolled out as a separate document from the requirements, gain approval and baseline the Scope document before proceeding.</td></tr>
</table></td></tr>
</table>

# Section 3: Document the Requirements (of the system-of-interest)

**Instructions:** The actions defined in the boxes below are performed in parallel and iteratively. When the flow has been completed for portions of the Requirements Document, those portions can be moved on to Section 4 of this procedure.




**Inputs/Entry Criteria:**

- A validated and approved Scope for this system-of-interest.
- The authors and system/lead engineers have been trained on how to develop and document good requirements, i.e., NET REQ, REQ-T class [2] or equivalent training which covers the same phases as this procedure.

**Outputs/Exit Criteria:**

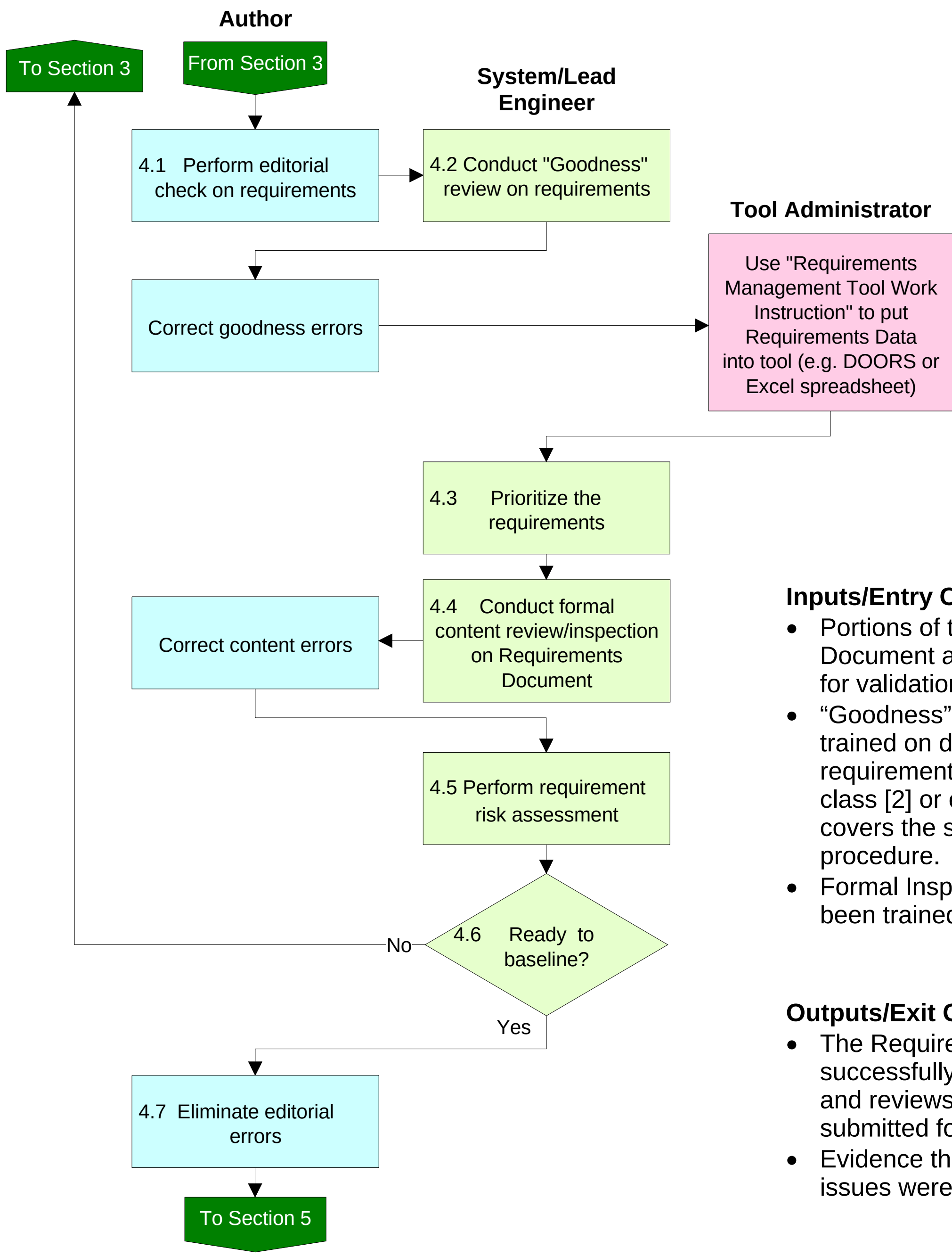
- Portions of the Requirements Document are completed and ready for validation.

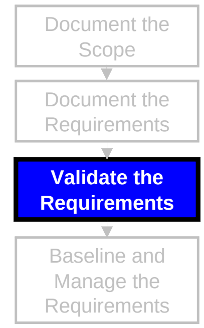
<table>
<tr><th>STEP</th><th>ACTION TO BE TAKEN</th></tr>
<tr><td>3.1</td><td>Collect and document the requirements<br>a. Become familiar with system-of-interest parent documents and the Scope for this system-of-interest.<br>b. With the Scope of this system-of-interest as a starting point, and with input from the stakeholders, collect and document each individual requirement for this system-of-interest following the Rules for Writing Good Requirements (Appendix A, A.1 - A.4).<br><u>GUIDANCE</u>: Requirements Documents should only contain product requirements. Project requirements are generally captured in a Statement of Work (SOW) or project plans.</td></tr>
<tr><td>3.2</td><td>Document the rationale for each requirement<br>As each requirement is documented, record (per the project’s template) the rationale, which includes of the following items:<br>• The reason for the requirement (i.e., why requirement exists and the source of the requirement).<br><u>GUIDANCE</u>: Often the reason for the requirement is not obvious and it may be lost if not recorded as the requirement is being documented. The reason may point to a constraint, trade or design study, or operations concept. If there is a “traceability link” from a higher-level requirement that completely explains the reason for the requirement, then simply reference the link.<br>• Assumptions made while developing the requirement. Assumptions must be confirmed before the requirements can be baselined.<br>• The relationships with the product’s expected operations (e.g., expectations about how customers will use a product).<br><u>GUIDANCE</u>: This may be done with a link to the Operational Concept.<br>• High-level design choices that drive low-level requirements (e.g., trade study results).<br><u>GUIDANCE</u>: If the requirement states a method of implementation, the rationale must state why the solution is being limited to this one method of implementation.<br>When a Requirements Document accompanies a contract or task order, the rationale is not part of the contractual binding language. The Requirements Document and rationale may accompany the Request for Proposals.</td></tr>
<tr><td>3.3</td><td>Ensure that the requirements are at the correct hierarchical Requirements Document level and are properly allocated<br>a. Choose the correct architectural level, i.e., system-of-interest, for documenting requirements. Use the following questions to aid in determining if the requirements are at the correct Requirements Document level. If the answer to any of these questions is no, the requirement is likely to be at the incorrect level.<br>• Does “why do we need the requirement” take you back directly to the level above?<br>• Does the requirement allow you more than one architecture or design option for the next level?<br>• Does it make sense to verify the requirement at this level?<br>• Have all constraints that apply to this level been captured?<br>b. After preliminary decomposition of the system-of-interest into lower-level systems of interest, document (per the project’s template) the allocation of each requirement at this level to the next lower-level system-of-interest that must accomplish the requirement. This allocation is necessary so that the author at the next level knows exactly which parent-level requirements apply.<br>c. Use the questions below to help verify the completeness and correctness of the requirements allocations.<br>• Is every requirement allocated?<br>• Are there any duplicate or conflicting requirements from parent source Requirements Documents?<br>• Can the system-of-interest to which you have allocated the requirement entirely satisfy the requirement by itself? Or should it be allocated to more than one system-of-interest?<br>• Are interfaces simple and controllable? (If not, this may indicate a potential architectural problem.)</td></tr>
</table>

| STEP | ACTION TO BE TAKEN |
|---|---|
| **3.4** | **Document requirements traceability**<br>a. As each requirement is documented, record (per the project's template) its lineage/traceability to at least one parent at the next higher level.<br>b. Use the following checklist to validate the documented traceability. (It is preferable to have someone other than the author perform this activity.)<br>• Are you able to trace each requirement back to requirements (or Scope, for the top-level requirements) in the level above it and vice versa? The requirement should be evaluated to assure that the requirements trace is correct and that it fully answers the parent requirements. If it does not, some other requirement(s) must complete fulfillment of the parent requirement.<br>• If there is no parent, is the requirement "gold plating" or is there a missing requirement at the higher level?<br>• Are you able to resolve duplication between levels? If a requirement is simply repeated at a lower level and it is not an externally imposed constraint, perhaps the requirement does not belong at the higher level.<br>c. The author corrects all traceability errors. |
| **3.5** | **Document how each requirement will be verified**<br>a. For each requirement, document the verification method(s) (i.e., inspection, demonstration, analysis, or test). If a requirement is unverifiable, it must be rewritten. In most cases, the contractor is required to document the verification methods (e.g., verification matrix) and provide them to the government as a deliverable. There may be some exceptions to this such as specific verification methods that the government may have to impose (e.g., federal regulations, national consensus codes, national and international standards requiring explicit test or inspection methods). The government requirements are part of the contractual binding language.<br>b. If the effort will be performed in-house (not on contract), for each requirement, document the verification level and phase.<br>GUIDANCE: Levels of verification are, for example, component, subsystem, and system. Phases of verification are, for example, design, manufacture, verification.<br>c. Document any new/additional requirements that are uncovered during determination of the verification method (e.g., extra connectors on the wiring harness to connect to test instrumentation or external power, extra data on a display or in a database to give visibility into an internal process). |
| **3.6** | **Maintain the Document Tree**<br>Use the following checklist to aid in developing and maintaining the Document Tree.<br>• Segregate requirements into documents of manageable size and along organizational management lines.<br>• Segregate requirements that may be revised frequently into separate documents.<br>• Segregate requirements that will be contractually binding to each outside party, and create separate Requirements Documents for each party.<br>• Identify approval levels for the separate Requirements Documents. |

# Section 4: Validate the Requirements (of the system-of-interest)

**Instructions:** The actions defined in the boxes below are performed sequentially.




**Inputs/Entry Criteria:**

- Portions of the Requirements Document are completed and ready for validation.
- "Goodness" reviewers have been trained on documenting good requirements, i.e., NET REQ, REQ-T class [2] or equivalent training which covers the same phases as this procedure.
- Formal Inspections participants have been trained on the method.

**Outputs/Exit Criteria:**

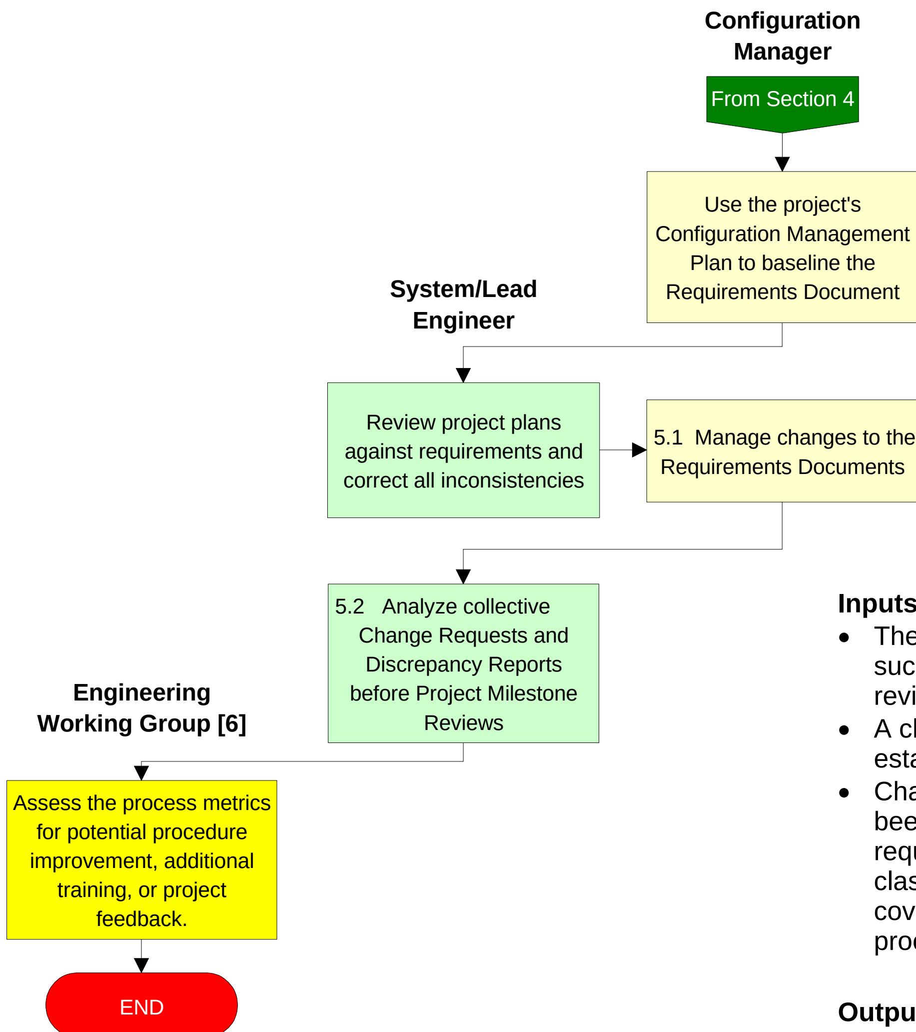
- The Requirements Document has successfully completed all inspections and reviews and is ready to be submitted for baseline.
- Evidence that review / inspection issues were resolved

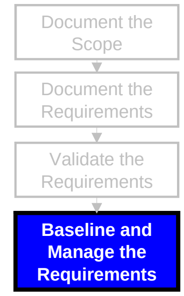
| STEP | ACTION TO BE TAKEN |
|---|---|
| **4.1** | **Perform editorial check on requirements**<br>a. Perform an editorial check using the Editorial Checklist, A.2 in Appendix A. Use “redlines” so the author can check to see if the change would corrupt the meaning.<br>GUIDANCE: This check should be performed on sections or portions of the document as they become available, to catch systemic problems early and increase efficiency. Consider having someone other than the author perform this activity.<br>b. The author corrects all editorial errors. |
| **4.2** | **Conduct “Goodness” review on requirements**<br>a. Conduct a “Goodness” review using the Goodness Checklist, A.3 in Appendix A. This review is conducted with 2-3 trained technical staff.<br>GUIDANCE: This check should be performed on sections or portions of the document, as they become available, to catch systemic problems early and increase efficiency.<br>b. Provide redlines/review comments back to the author for disposition.<br>c. Ensure document has been through editorial and goodness review before proceeding to step 4.4. |
| **4.3** | **Prioritize the requirements**<br>a. Determine when to prioritize the requirements. Prioritization must be performed before design.<br>GUIDANCE: Prioritization can be performed now (i.e., to take advantage of input on the priorities from the stakeholders during the upcoming review) or performed after the content review (i.e., deferred until a firmer set of requirements is established). Prioritize before design to guide architecture, design tradeoffs, system “builds”, and prototyping; enable effective downstream descopes due to schedule and resource shortfalls; and manage requirement additions and risk.<br>b. Define the schema that will be used to prioritize the requirements.<br>GUIDANCE: A numbering scheme of 1-2-3 may be used, where 1 = essential, mandatory, nonnegotiable (i.e., minimum to satisfy customer need, not susceptible to trades), or urgent requirements; 2 = useful (enhance the product value to the customer, these could be written as a “should”), negotiable (subject to trade), or slightly deferrable requirements; and 3 = desirable if low cost, flexible, longer delay requirement or can be readily descoped.<br>c. With input from all the stakeholders (including the developers), document each requirement’s priority.<br>GUIDANCE: Ask all stakeholders to classify the requirements by priority. Use the operational scenarios to help in the classification. Often, it is easiest to identify the 1’s and 3’s first and allow everything else to default to 2.<br>d. Resolve the priority differences between stakeholders.<br>e. Maintain the priorities throughout development.<br>GUIDANCE: Prioritization is not finished until the last version of the product is done. Priorities are most likely to change when there are major budget or schedule changes; when new information exists about cost, schedule, or technical feasibility; when external interfaces change; or when the designers want a priority change to match design. Also, when the customer brings new requirements that require deferring some existing requirement, reassess priorities to make certain that the least important priorities are the ones being deferred. |
| **4.4** | **Conduct formal content review/inspection on Requirements Document**<br>a. Determine whether to perform a formal review or Formal Inspection [4].<br>GUIDANCE: Formal reviews differ from Formal Inspection in that (1) formal reviews generally cover the entire document while Formal Inspections cover only a limited number of pages per inspection; (2) Formal Inspections identify defects (Inspectors may provide suggestions outside of the inspection meeting.) while formal reviews can require participants to both identify defects and provide recommended corrections; and (3) defects are dispositioned as part of the Formal Inspection meeting, while formal reviews disposition both the defects and the suggested corrections outside of the meeting.<br>b. Include representatives from all stakeholders in the review/inspection. Also, consider inviting outside experts for their project or domain experience.<br>GUIDANCE: Reviewers experienced in different lifecycle phases and aspects of use will be best equipped to find the omissions and incorrect facts in those phases and aspects. Include reviewers from developers, maintainers, manufacturers, testers, installers, and especially users.<br>c. Educate your review team on what constitutes a good requirement.<br>GUIDANCE: Assign the reading of Chapter 7, pages 101-119 of reference [5] to reviewers who have not already had training on writing good requirements. |
| | d. Perform the review. |

<table>
<tr><th>STEP</th><th>ACTION TO BE TAKEN</th></tr>
<tr><td></td><td>Conduct review or inspection following associated instructions from Appendix B.<br><table><tr><th>If</th><th>Then</th></tr><tr><td>Formal review</td><td>Follow Formal Review Instructions in B.1.</td></tr><tr><td>Formal Inspection</td><td>Follow Formal Inspection Instructions in B.2.</td></tr></table></td></tr>
<tr><td>4.5</td><td>Perform requirement risk assessment.<br>a. Address requirements volatility risks:<br>1. Identify volatile requirements.<br>2. Modify sensitive requirements to eliminate the need to change the requirement if the volatility becomes reality or to add additional requirements to make the design better able to accommodate the volatility.<br>3. Develop a plan to manage the development effort through requirement volatility when eliminating sensitivity is not possible.<br><u>GUIDANCE</u>: If the requirements are volatile (i.e., likely to change) consider assigning a stability rating to each requirement. For example: use a scheme of ‘high-medium-low’ where high = it is highly likely that the requirement will change; medium = the requirements may change; and low = the requirement is very unlikely to change.<br>b. Identify technical feasibility risks of each requirement and, where possible, modify requirements to reduce the risk and stay consistent with budget and schedule.<br>c. Assess the schedule and budget adequacy for the set of requirements to be included in the baseline.<br><u>GUIDANCE</u>: Identified risks should be addressed and mitigated according to the project’s Risk Management Plan. Refer to pages 193-198 of reference [5] for additional information on how to reduce requirements risks.</td></tr>
<tr><td>4.6</td><td>Ready to baseline?<br>Decide if ready to baseline the Requirements Document.<br><table><tr><th>If</th><th>Then</th></tr><tr><td>High risk</td><td>Postpone the Requirements Document baselining until the risk can be mitigated (i.e., repeat the associated steps above to reduce the risk).</td></tr><tr><td>Low risk</td><td>Proceed to the next step to prepare the Requirements Document for baselining.</td></tr></table></td></tr>
<tr><td>4.7</td><td>Eliminate editorial errors<br>a. Use the Editorial Checklist, A.2 in Appendix A, to ensure that each requirement is editorially correct. (Consider having someone other than the author perform this activity. Use “redlines” so the author can check to see if the change would corrupt the meaning.)<br><u>GUIDANCE</u>: As requirement defects discovered during the review/inspection are corrected, editorial problems may inadvertently result. A final check before baselining can reduce formal changes later.<br>b. The author corrects all editorial errors.<br>c. Submit the Requirements Document for baseline.</td></tr>
</table>

## Section 5: Baseline and Manage Requirements (of the system-of-interest)

**Instructions:** The actions defined in the boxes below are performed throughout the project.




**Inputs/Entry Criteria:**

- The Requirements Document has successfully completed all the reviews and is ready to be baselined.
- A change control process has been established for the project.
- Change Request evaluators have been trained on documenting good requirements, i.e., NET REQ, REQ-T class [2] or equivalent training which covers the same phases as this procedure.

**Outputs/Exit Criteria:**

- The Requirements Document is baselined and under configuration management.
- Change Requests have been dispositioned.
- Process metrics have been collected and analyzed.

<table>
<tr><th>STEP</th><th>ACTION TO BE TAKEN</th></tr>
<tr><td>5.1</td><td>Manage changes to the Requirements Documents<br>a. Using the project’s Change Request (CR) form, document the requested requirements statement(s) change with justification. Note: All requirements must include a rationale, allocation, traceability, and verification method.<br>b. Evaluate new or modified requirements against the checklists A.1 - A.3 and applicable items of A.4 in Appendix A. Work with the submitter to correct any deficiencies identified.<br>c. Perform a thorough change impact assessment from the standpoint of all relevant stakeholders. Include impacts on parent and child requirements, existing commitments, cost, and schedule. Communicate potential changes to all stakeholders who are impacted.<br>d. Using the project’s Configuration Management process, approve or disapprove each change.<br>e. For approved changes:<br>• Distribute the changes to all who are impacted<br>• Maintain the requirements change history<br>• Make approved change to the Requirements Document<br>• Ensure all project plans, and other affected work products are updated to reflect the changes</td></tr>
<tr><td>5.2</td><td>Analyze collective Change Requests and Discrepancy Reports before Project Milestone Reviews<br>a. At the end of each development phase (e.g., preliminary design, detailed design, low-level testing, system-level qualification/acceptance testing), review CRs to identify nontrivial requirement changes (e.g., changes that affected design, required rework, impacted commitments, cost, or schedule). Sort those changes into three categories:<br>• Modified<br>• Added<br>• Deleted<br>GUIDANCE: For space flight projects, changes that affect cost, schedule, performance, interface, or other project defined criteria are referred to as Class I [7].<br>b. Analyze these CRs to determine the cause of the changes and if improvements can be made to eliminate further changes. Provide feedback to this LMS CP author if the analysis indicates changes/improvements in this procedure are warranted.<br>c. At the end of each development phase, review product Discrepancy Reports (DRs) to identify those that involve requirements problems (i.e., separate problems in meeting the requirements from problems in the requirements documents). Sort DRs involving requirements problems into three categories:<br>• Incorrect requirement<br>• Ambiguous or misunderstood requirement<br>• Missing requirement<br>GUIDANCE: For space flight projects, DRs are referred to as Nonconformance-Failure Reports [7].<br>d. Record the following metrics:<br>• Name of the project and the system-of-interest<br>• Phase currently completing<br>• Date<br>• Total number of requirements<br>• Number of requirement changes in each of the three categories (Modified, Added, Deleted)<br>• Total number of DRs involving requirements in each of the three categories (Incorrect, Ambiguous, Missing)<br>GUIDANCE: The Requirements Metrics Collection Worksheet (See [8] for sample.) can be used to record the metrics.<br>e. Present at project milestone reviews (e.g., Preliminary Design Review, Critical Design Review, Test Readiness Review, and System Acceptance Review) the percent of requirements changes in each category, the results of the CR analysis, and the percent of discrepancies related to requirements problems in each category.<br>GUIDANCE: This information will provide the reviewers with an indication of the stability and quality of the requirements, the associated risks, and readiness to proceed to the next phase.</td></tr>
</table>

# Appendix A: Rules for Writing Good Requirements

## A.1 Use of Correct Terms

**Shall** = requirement; **Will** = facts or declaration of purpose; and **Should** = goal.

## A.2 Editorial Checklist

- The requirement is in the form “product ABC shall XYZ”. A requirement must state “The product shall (do, perform, provide, weigh, or other verb followed by a description of what), i.e., must be in “**Who**” shall “**What**” form; uses active rather than passive voice.
  Example Product requirements:
    - The system shall operate at a power level of…
    - The software shall acquire data from the…
    - The structure shall withstand loads of…
    - The hardware shall have a mass of…
- The requirement uses consistent terminology to refer to the product and its lower-level entities.
- The requirement is grammatically correct.
- The requirement is free of typos, misspellings, and punctuation errors.
- The requirement complies with the project’s template and style rules.

## A.3 Goodness Checklist

Is each requirement…

- Clear and understandable?
  - Can only be understood one way?
  - Free from indefinite pronouns (this, these)?
  - Expressing only one thought per requirement statement? A standalone statement (as opposed to multiple requirements in a single statement or a paragraph that contains both requirements and rationale)?
  - Stated simply and concisely?
  - Stated positively (as opposed to negatively, e.g., “shall not”)?
- Free of ambiguities (e.g., as appropriate, etc., and/or, support, but not limited to, be able to, be capable of)?
- Free of unverifiable terms (e.g., flexible, easy, sufficient, safe, ad hoc, adequate, accommodate, user-friendly, useable, when required, if required, appropriate, fast, portable, light-weight, small, large, maximize, minimize, sufficient, robust, quickly, easily, clearly, other ”ly” words, other “ize” words)?
- Free of implementation? (Requirements should state WHAT is needed, NOT HOW to provide it, i.e., state the problem not the solution. Ask, “Why do you need the requirement?” The answer may point to the real requirement.)
- Free of descriptions of operations? (Don’t mix operation with requirements; update the Operational Concept instead. To distinguish between operations and requirements ask the questions: “Does the developer have control over this? Is this a need the product must satisfy or an activity involving the product?” Sentences like “The operator shall…” are almost always operational statements not requirements.)
- Free of “To Be Determined” (TBD) values? (A best guess, marked To Be Resolved (TBR), with the rationale should replace these.)
- Complete with tolerances for qualitative/performance values (e.g., Less than, greater than or equal to, plus or minus, 3 sigma root sum squares)?
- Accompanied by intelligible rationale, including any assumptions? Can you validate (Do I concur with) the assumptions? Assumptions must be confirmed before baselining.
- Traceable to requirements (or to Scope, for the top-level requirements) in the level above it?
- Identified with a verification method(s) (i.e., test, demonstration, analysis, or inspection)? Does a means exist to measure its accomplishment? Can you state the criteria required for verification? Can compliance be verified?
- Located in the proper section of the document?
- Defined at the correct level?
- Unique (as opposed to redundant)?
- Consistent with other requirements (as opposed to conflicting)?

## A.4 Content Review / Inspection Checklist

CLARITY
1. Are the requirements clear and unambiguous? (i.e., Are there aspects of the requirement that are not understood; can the requirement be misinterpreted?)
2. Are the requirements concise and simple?

COMPLETENESS
1. Are requirements stated as completely as possible? Have all incomplete requirements been captured as TBRs?
2. Are any requirements missing? For example have any of the following requirements areas been overlooked: functional, performance, interface, environment (development, manufacturing, test, transport, storage, operations), facility (manufacturing, test, storage, operations), transportation (among areas for manufacturing, assembling, delivery points, within storage facilities, loading), training, personnel, operability, safety, security, appearance and physical characteristics, and design.
3. Have all assumptions been explicitly stated?

COMPLIANCE
1. Are all requirements at the correct level (e.g., system, segment, element, subsystem)?
2. Are requirements specified in an implementation-free way so as not to obscure the original requirements, i.e., do the requirements state “what” and not “how”?
3. Are requirements specified in an operations-free way? Is this a requirement the developer has control over, something the product must do, or a quality it must have, rather than an activity involving the product?

CONSISTENCY
1. Are the requirements stated consistently without contradicting themselves or the requirements of related systems?
2. Is the terminology consistent with the user and sponsor’s terminology? With the project glossary?
3. Is the terminology consistently used through out the document?
4. Are the key terms included in the project’s glossary?

TRACEABILITY
1. Are all requirements needed? Is each requirement necessary to meet the parent requirement? Is each requirement a needed function or characteristic? Distinguish between needs and wants. If it is not necessary, it is not a requirement. Ask, “What is the worst that could happen if the requirement was not included?”
2. Are all requirements (functions, structures, and constraints) traced to mission or system-of-interest Scope (i.e., need(s), goals, objectives, constraints, or operational concept)?
3. Is each requirement stated in such a manner that it can be uniquely referenced in subordinate documents?
4. Is allocation to the next lower level documented?

CORRECTNESS
1. Is each requirement correct?
2. Is each stated assumption correct? Assumptions must be confirmed before the document can be baselined.
3. Are the requirements technically feasible?

FUNCTIONALITY
1. Are all described functions necessary and together sufficient to meet mission and system goals and objectives?

PERFORMANCE
1. Are all required performance specifications and margins listed? (E.g., consider timing, throughput, storage size, latency, accuracy and precision).
2. Is each performance requirement realistic?
3. Are the tolerances overly tight? Are the tolerances defendable and cost-effective? Ask, “What is the worst thing that could happen if the tolerance was doubled or tripled?”

INTERFACES

1. Are all external interfaces clearly defined?
2. Are all internal interfaces clearly defined?
3. Are all interfaces necessary, sufficient, and consistent with each other?

MAINTAINABILITY

1. Have the requirements for system maintainability been specified in a measurable, verifiable manner?
2. Are requirements written so that ripple effects from changes are minimized (i.e., requirements are as weakly coupled as possible)?

RELIABILITY

1. Are clearly defined, measurable, and verifiable reliability requirements specified?
2. Are there error detection, reporting, handling, and recovery requirements?
3. Are undesired events (e.g., single event upset, data loss or scrambling, operator error) considered and their required responses specified?
4. Have assumptions about the intended sequence of functions been stated? Are these sequences required?
5. Do these requirements adequately address the survivability after a software or hardware fault of the system from the point of view of hardware, software, operations personnel and procedures?

VERIFIABILITY / TESTABILITY

1. Can the system be tested, demonstrated, inspected, or analyzed to show that it satisfies requirements?
2. Are the requirements stated precisely to facilitate specification of system test success criteria and requirements?

DATA USAGE

1. Where applicable, are "don't care" conditions truly "don't care"? ("Don't care" values identify cases when the value of a condition or flag is irrelevant, even though the value may be important for other cases.) Are "don't care" conditions values explicitly stated? (Correct identification of "don't care" values may improve a design's portability.)

# Appendix B: Formal Review/Inspection Instructions

## B.1 Formal Review Instructions

1. Prepare the review package, which includes the document to be reviewed, background materials, instructions for the review, and Review Item Disposition (RID) form (see [8] for sample). Instructions include the following: a) read the entire package before writing any RIDs (including the background material such as scope and operational concepts), b) use Content Review/Inspection Checklist, A.4 in Appendix A, to aid in identifying errors, c) document comments and corrections on the provided RID form, d) return completed RIDs in electronic form.
   GUIDANCE: At a minimum the RID form should require: (a) the original requirement (unless missing), (b) description of problem, (c) recommended new requirement or requirement/document change, and (d) justification (why the change is needed).
2. Conduct an overview meeting with the review team. At this meeting, review the scope and operational concepts of the system-of-interest. Provide the reviewers with the review package and brief them on the contents, the type of comments you want, the schedule, and method of submitting the RIDs.
3. Obtain completed RIDs.
4. Disposition the RIDs.
   a. Sort the recommendations into three groups: (1) definitely accept (or accept with modifications) (2) maybe accept; and (3) definitely not accept. Use the following checklist to help determine if a recommendation should be accepted.
      Acceptance Criteria Checklist:
      - Is the recommended addition within scope?
      - Is the recommended addition necessary to meet the scope?
      - Does the recommended change correct an error or assumption?
      - Does the recommended change address technical feasibility or constraints?
      - Does the recommended change clarify a requirement that can be misinterpreted?
      - Is the recommended deletion out of scope, gold plating, or duplicate?
   b. For the second group, conduct meetings with the reviewers, authors, and key stakeholders to resolve issues; and, for the third group, document the reasons for disapproval.
   c. For those RIDs that have been accepted, evaluate new or modified requirements against the checklists A.1 - A.3 and applicable items of A.4 in Appendix A. Work with the submitter to correct any deficiencies identified.
   d. Incorporate acceptable reviewer recommendations.
5. Communicate and Revise the Disposition.
   a. Circulate to all review participants a review recommendation summary showing the disposition of all RIDs.
   b. Hold a meeting with the reviewers to provide them an opportunity to comment on the recommendation summary.
6. Revise the requirements based on the outcome of the meeting.

## B.2 Formal Inspection Instructions

Conduct the Formal Inspection following the instructions defined in the Instructional Handbook for Formal Inspections [4]. In addition, add the following to the planning stage activities of the handbook:

a. Include completion of the following three activities to the Formal Inspection entrance criteria:
   - Perform terminology check (A.1)
   - Perform editorial check (A.2)
   - Perform goodness check (A.3)
b. Include Content Review/Inspection Checklist A.4 in Appendix A in the inspection package that is distributed.

# Appendix C: Acronyms, Definitions, and References

## C.1 Acronyms

**CR –** Change Request
**DR –** Discrepancy Report
**ICD** – Interface Control document
**IRD** – Interface Requirements Document
**K** – Thousand
**M** – Million
**NET** – NASA Engineering Training
**NET REQ –** NET System Requirements
**NET REQ–T –** NET System Requirements –Team
**RID –** Review Item Disposition
**SOW** – Statement of Work
**TBD** – To be determined
**TBR** – To be resolved

## C.2 Definitions

**Constraints:** External items the project cannot control that must be met (e.g., regulations, work on existing systems, the product is needed by a certain date). During requirements development, constraints evolve into requirements or are used as the rationale for the requirements.[1]

**Interface Control Document (ICD):** Defines the interface for existing products. [5, p.87]

**Interface Requirements Document (IRD):** Defines the interface requirements for products that do not exist yet and must be developed. [5, p.87]

**Change Request (CR):** Generic term referring to the documentation used to submit and disposition requested changes to baseline documents. [1]

**Project Milestone Reviews:** The lifecycle series of rigorous system-level technical and programmatic evaluations conducted at key formulation and implementation milestones. Key milestones in this context are the major transition points in the lifecycle, such as the transition from requirements development to design activities, final design to manufacturing, and the transition from the assembly and integration of components to system-level environmental testing. Project Milestone Reviews may include, but is not limited to, System Concept Review, Requirements Review, Preliminary Design Review, Critical Design Review, Pre-Environmental Review, Test Readiness Review, Pre-Ship Review, and Operational Readiness Review. [9]

**Discrepancy Report (DR):** Generic term referring to the documentation used to officially report discrepancies or nonconformance resulting from verification and validation of products or systems. [1]

**Phases of the Lifecycle:** Formulation, design, development, manufacturing, verification, validation, shipping, storage, installation, pre-deployment, deployment, training, operations, maintenance, upgrading, disposal. [1]

**Product Requirement:** A single statement of something the product must do or a quality the product must have. [1]

**Project Requirement:** A single statement of a task to be done or what the provider will deliver. [1]

**Requirements Document:** See definition for Requirements Specification. (Note: The terms "Requirements Document" and "Requirements Specification" are used synonymously.)

**Requirements Specification:** A document that specifies the requirements for a system *[the system-of-interest]* or component. Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards. [10]

**Scope (of the system-of-interest):** The need(s), goals and objectives, constraints (including budget and schedule), operational concepts, external interfaces, and associated assumptions. [1]

**Stakeholders:** Anyone who has a vested interest in the project (i.e., systems engineering, electrical engineering, mechanical engineering, software engineering, manufacturing, testing, handlers of packaging, storing, shipping, and disposal, trainers, users during training, operations, and upgrades, operations, maintenance, and logistics). [1]

---

[1] Definition is from course materials of NASA Engineering Training class "Systems Requirements" (REQ) after June 14, 2005. See [2] for course information and class schedule.

A group or individual that is affected by or in some way accountable for the outcome of an undertaking. Stakeholders may include project members, suppliers, customers, end users, and others. [11]

**System-of-interest:** The definition of a particular system- of-interest to be engineered depends on the practitioner's responsibilities, scope of assignment, and interest. For example, within a hierarchy of systems, one person's system-of-interest may be viewed as an element in another person's higher-level system-of-interest. [9]

**Validation (of requirements):** The process of confirming the completeness and correctness of the requirements [5, p.157]. Validation is performed after requirements or changes to them are defined.

**Verification (of system-of-interest):** Verification is a process (consisting of tests, inspections, demonstrations, and analysis) of confirming that the designed and built product meets the requirements. [5]

## C.3 References

1 NASA SP-6105, NASA Systems Engineering Handbook, June 1995. (URL: http://ldcm.gsfc.nasa.gov/library/Systems_Engineering_Handbook.pdf)

2 NASA Engineering Training - NASA training Master Schedule (URL: http://appl.nasa.gov/businessunits/career/classroom/calendar.html)

3 IEEE Standard 1362-1998, IEEE Guide for Information Technology – System Definition – Concept of Operations (ConOps) Document (available for government use at URL: http://ieeexplore.ieee.org/xpl/standards.jsp).

4 Instructional Handbook for Formal Inspections (available at URL: <http://sw-eng.larc.nasa.gov/process/docslistnew.cfm>).

5 Customer Centered Products: Creating Successful Products Through Smart Requirements Management, I. F. Hooks and K. A. Farry, AMACOM (2001), ISBN 0-8144-0568-1.

6 Langley Policy Directive (LAPD) 1150.2, Boards, Panels, Committees, Councils and Teams (available from LMS at http://lms-r.larc.nasa.gov/policy_directives.cfm).

7 Langley Procedural Requirements (LPR) 5300.1, Space Product Assurance (available from LMS at http://lms-r.larc.nasa.gov/procedure_guideline.cfm).

8 Building Quality Product Requirements Web Site, URL: http://sw-eng.larc.nasa.gov/docs/requirements_capture_and_management.html

9 NASA Procedural Requirements (NPR) 7120.5, NASA Program and Project Management Processes and Requirements (available from LMS or NODIS at URL: http://nodis3.gsfc.nasa.gov).

10 IEEE Standard 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology (available for government use at URL: http://ieeexplore.ieee.org/xpl/standards.jsp).

11 Capability Maturity Model Integration (CMMI) for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing; Staged Representation; Version 1.1, Software Engineering Institute (2002), Technical Report CMU/SEI-2002-TR-012 (URL: http://www.sei.cmu.edu/cmmi/models/models.html).

# Appendix D: Criteria[2] for Procedure Applicability.

A project's control class is determined as follows. Assess the potential risks of the project using the Classification Criteria in the left column of Table D-1. Where the project meets the criteria identified on the left, assign the corresponding Project Class from the right side. If more than one class is indicated, then assign the highest (leftmost) class to the project. For the project to be Minimal-Control class, *all* of the following must be true:

- The project work products will have no effect on products or services outside the Project Team, other than publication of results clearly identified as research.
- There is no intent to use or distribute the project work products outside the Project Team or Requestor.
- The highest class determined by Table D-1 is Minimal Control.

## Table D-1: Criteria for Selection of Project Class

| Classification Criteria | Project Classes | | | |
|---|---|---|---|---|
| | Critical Control | High Control | Low Control | Minimal Control |
| Potential for safety mishap: | | | | |
| Loss of life [a] | X | | | |
| Serious injury [b] | X | | | |
| Damage to equipment[3] greater than $1M [c] | X | | | |
| Potential for mission failure: | | | | |
| Catastrophic or partial mission failure [d,e] | | X | | |
| Potential for waste of personnel resource investment: [f] | | | | |
| 8 or more Full Time Equivalents (FTE) on projects | | X | | |
| More than 2 FTE and less than 8 FTE on projects | | | X | |
| 2 or less FTE on projects | | | | X |
| Potential for waste of facility resources: [g] | | | | |
| $250K or more | | X | | |
| Less than $250K and greater than $50K | | | X | |
| $50K or less | | | | X |
| Potential for adverse publicity: [h] | | | | |
| At the National / Agency / Center level | | X | | |
| At the implementing Directorate / Branch level | | | X | |
| Unlikely at the Branch or higher level | | | | X |
| Potential effect on routine operations: [i] | | | | |
| Center inconvenience or facility work stoppage | | X | | |
| No more than a facility inconvenience | | | X | |

a. **Potential for loss of life.** Is the product the primary means of controlling or monitoring systems that have the potential to cause the death of an operator, crew member, support staff, or bystander? The presence of manual overrides and failsafe devices is not to be considered. Examples of products with the potential for loss of life include:
   (1) Flight and launch control software for human-rated systems,
   (2) Products controlling life support functions,
   (3) Products controlling hazardous materials with the potential for exposing humans to a lethal dose,
   (4) Products controlling mechanical equipment (including vehicles) that could cause death through impact, crushing, or cutting,
   (5) Any product that provides information to operators where an inaccuracy could result in death through an incorrect decision (e.g., mission control room displays).

[2] Criteria taken from Draft NASA Software Assurance Standard
[3] LPR 1740.4, Facilities Systems Safety Analysis and Configuration Management, (URL: http://lms-r.larc.nasa.gov)

b. **Potential for serious injury.** Serious injury is defined as loss of use of digit or limb, or sight in one or both eyes, hearing, or exposure to substance or radiation that could result in long term illness. This rating considers only those cases where the product is the primary mechanism for controlling or monitoring the system. The presence of manual overrides and failsafe devices is not to be considered. Examples of products with potential for serious injury include products controlling milling or cutting equipment, class IV lasers, or X-ray equipment.

c. **Potential for damage to equipment.** This is a measure of the cost (in dollars) of physical resources that are placed at risk of damage, destruction, or loss due to a product failure. Potential collateral damage is to be included. This is exclusive of mission failure. Examples include the following:
   (1) Damage to the Shuttle robotic arm due to the premature firing of a payload's thrusters,
   (2) Damage to a wind tunnel drive shaft due to a sudden change in rotation speed.

d. **Potential for catastrophic mission failure.** Can a problem in the product result in a catastrophic failure of the mission?[4] Products controlling navigation, communications, or other critical systems whose failure would result in loss of vehicle or total inability to meet mission objectives would fall into this category.

e. **Potential for partial mission failure.** Can a problem in the product result in a failure to meet one or more of the overall mission[3] objectives? Examples of this category include a product controlling one of several data collection systems or a product supporting a given experiment which is not the primary purpose of the mission.

f. **Potential for waste of personnel resource investment.** This is a measure or projection of the effort (in work-years: civil service, contractor, and other) invested in the project. The measure of effort includes all project lifecycle phases (e.g., planning, design, maintenance). This shows the level of effort that could potentially be wasted if the project does not meet requirements.

g. **Potential for waste of facility resources.** This is a measure of the cost (in dollars) of consumable resources and/or operational costs that are placed at risk of waste due to a product failure. An example is resources consumed (electricity, liquid nitrogen) during a research facility test that has to be re-run due to a product failure.

h. **Potential for adverse publicity.** This is a measure of the potential for negative political and public image impacts stemming from a failure of the system as a result of product failure. The unit of measure is the geographical or political level at which the failure will be common knowledge. The potential for adverse publicity is evaluated based on the history of similar efforts.

i. **Potential effect on routine operations.** This is a measure of the potential to interrupt business. There are two major components of this rating factor: scope and impact. Scope refers to who is affected. The choices are facility or Center. The choices for impact are inconvenience and work stoppage. Examples include the following:
   (1) A faulty firewall that failed to protect against a virus resulting in a 4-hour loss of e-mail capabilities at a Center would be a "Center inconvenience."
   (2) If data acquisition software failed but a workaround was possible, the test would not be lost but the participants would suffer a "facility inconvenience." However, extended loss of access to a system would be a "facility work stoppage."

[4] For the purposes of this document, a mission is a project required to follow NASA Procedural Requirements (NPR) 7120.5, NASA Program and Project Management Processes and Requirements.